The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage
VAST (2004)
Y. Chrysanthou, K. Cain, N. Silberman, F. Niccolucci (Editors)

As Time Flies By: Mixed Image and Model-Based Rendering of an Historical Landscape from Helicopter Images

M. Vergauwen[1], D. Pletinckx[3], G. Willems[1], F. Verbiest[1], L. Van Gool[1,2] & Truus Helsen[3]

[1] Kath. Universiteit Leuven, Dept. ESAT-PSI
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

[2] ETH Zürich D-ITET-BIWI
Gloriastrasse 35, CH-8092 Zürich, Switzerland

[3] Ename Center for Punblic Archaeology and Heritage Presentation
Abdijstraat 13-15, B-9700 Oudenaarde, Belgium

Abstract

For the preservation of cultural heritage to be successful the general public must be able to experience sites and reconstructions in an intuitive, yet convincingly real way. In this paper, a pipeline is discussed that can be employed to generate an interactive presentation of landscape reconstruction through a Quicktime VR object movie. Images of the current landscape are registered with virtual reconstructions of the same landscape through time.

Categories and Subject Descriptors (according to ACM CCS): I.4.5 [Image Processing and Computer Vision]: Reconstruction

1. Introduction

One very good strategy to get people interested in the past is by highlighting its link with the present. We discuss a way to let them explore the evolution of a landscape up till modern times. To that end, seamless transitions between previous states and current reality are made, based on a perfect registration of visualizations of virtual reconstructions of the past with morphed photographs of the present.

Many applications make use of the QuickTime technology provided by Apple, especially to create interactive applications through the QTVR suite [App]. One element of this technology are QTVR object movies. A QTVR object movie allows interactive access to a two-dimensional matrix of images. One of the ways to use this is to show an object from several different angles. The object may be shown in full rotation, often tilted at several angles as well. The images are arranged so that when the viewer drags them using the mouse, the object seems to tilt and rotate.

A more sophisticated use of the QTVR object technology is to build interactive 4-dimensional objects, by rotating the object through a horizontal cursor movement, and evolving the object through time through a vertical cursor movement. Through this technique one can show for example an opening flower in a 3-dimensional way. Since 1999, the Provincial Archaeological Museum of Ename has used 4-dimensional QTVR objects to show the evolution of the entire village over the last ten centuries [PSC02]. In this interactive TimeLine application, virtual models of 12 consecutive periods are visualized, and archaeological objects – on display in the same room – are linked by indicating their place where they where found in the appropriate historical period. In the TimeLine application a virtual camera spins at a fixed height around a fixed point, yielding 36 regularly spaced views on the geo-referenced virtual model, and this for all 12 historical periods. In this way, the columns of the 36×12 matrix of views show the evolution of the landscape from 1020 to 1780 AD from a specific point of view, with the columns evenly spaced on a predefined circle with a height of 150 m and a diameter of 260 m. The circle where the virtual camera was positioned is shown in fig. 1. This figure also shows the Ename site.

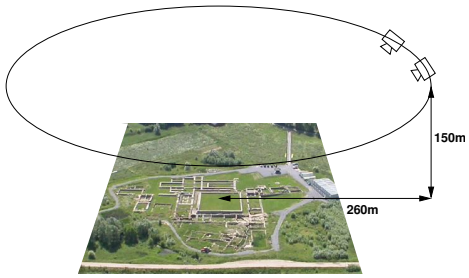

Figure 1: *The setup of the virtual cameras. They are distributed every 10 degrees on a circle with a radius of 260 meter at 150 meter altitude.*

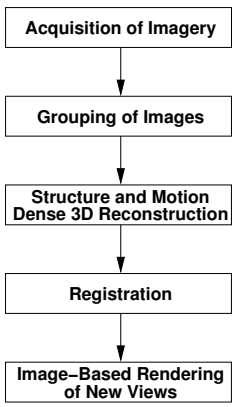

Figure 2: *Five steps in the pipeline will realize our final goal of making a QTVR object movie of the real site, registered with the virtual views of the existing TimeLine*

To extend this matrix to the present, as illustrated in fig. 3, we need to take images from exactly the same 36 viewpoints, and this at a height of 150 m. This is virtually impossible with straightforward photography however, and the paper proposes an alternative way, based on *image-based rendering*. The result is rewarding. In more than one way, this most recent, modern row of the TimeLine matrix (see the 4th row in fig. 3) is the most intriguing one, as it links the present – experienced by the visitor – to the past. Every element in the current landscape gets a meaning, a story, a reason why it is the way it is. By linking the past to the present, a general audience can be addressed by telling the story of how a place in time came to be.

2. Overview of the Pipeline

A processing pipeline has been devised to create modern views that are aligned with those already in the TimeLine matrix, as shown in fig. 2.

First step - In essence, this pipeline produces interpolated views, starting from a series of photographs of the site. Therefore, a first step consists of acquiring the latter. Since the virtual TimeLine images show the archaeological site from above, a helicopter was used to shoot the photographs. Section 4 explains this in more detail.

The use of a helicopter is only part of the solution. Even when combined with state-of-the art technology such as DGPS or steadycams, it is highly unlikely that one would manage to let any of the images coincide perfectly with one of the 36 target viewpoints as used in the TimeLine views. We have used Image-Based Rendering (IBR) for that purpose. IBR takes a series of close photographs taken from known directions and a 3D model of the scene as input, and then creates a kind of interpolated views, taking account of the 3D geometry. This may require that the camera is held by a robot and that laser scanners are used to provide the 3D geometry. IBR has therefore been confined to lab conditions at first. In the ESAT-PSI lab, we have developed a procedure that is able to compute both the camera calibration data and the dense 3D geometry directly from the photographs themselves [KPH*99]. Hence, images (that may be taken from unknown positions) are the sole input to this IBR system.

Second step - Typically IBR is applied based on images taken from many and widely spread directions, so as to be able to generate synthetic views from all around an object. Here, we are only interested in a very limited set of such synthetic views. Hence, we can actually split the task into a series of smaller IBR problems. For each target viewpoint images are selected that were taken in its vicinity. This image grouping step is explained in section 5.

Third step - In the previous step, the helicopter images have been subdivided into groups, where each group is to be used as the input to an IBR process for some of the target viewpoints. On average four target viewpoints could be rendered per selected group. For the images within each group, the camera parameters are extracted (calibration) and a 3D reconstruction is computed from their point of view, exploiting all the images in the group. The camera calibration includes both intrinsic (camera setting) and extrinsic (camera position and orientation) parameters. These processes - camera calibration and 3D geometry extraction are jointly referred to as *Structure and Motion (SaM)*. They are the subject of section 6.

Fourth step - In order to determine the actual target viewpoints in terms of the IBR images that have to be formed, the new images must be registered to the virtual target images. Registration is the process of bringing one coordinate frame, i.c. the frame of the reconstruction based on real images, to another coordinate frame, i.c. the frame of the virtually reconstructed site. In order to do so, correspondences between both frames are necessary. Since only few structures visible in the virtual TimeLine views are still easily recognizable in the real images of today, we opted for a manual process in which an archaeologist indicates points on the virtual recon-

Figure 3: *These 16 images are a small part of the* 36×13 *TimeLine image matrix, upgraded with present day images (last row), using methods described in this paper. Each column represents on of 36 different viewpoints (in this case arranged along a circular path), and the 12 first rows represent different eras from the past. The last row contains images from our own time, for the same viewpoints and in the form of photo-realistic images, so that visitors can directly relate to them. In this cutout of the complete TimeLine matrix only 4 eras are shown, namely 1020, 1250, 1780 and 2004.*

structions of the different eras and on the ortho-photo of the actual site. This process is described in section 7.

Fifth step - Finally, once data from all previous step have become available, newly rendered images can be produced using the IBR approach explained in section 8.

3. Contributions of this paper

Several novel techniques have been implemented to obtain the resulting QTVR matrix of fig. 3. Image-Based Rendering (IBR) techniques are used to generate the 36 requested views. IBR is typically used on data acquired in laboratory conditions. Some previous experience existed in bringing IBR outside the lab, using hand-held cameras instead. In this paper this approach has been lifted to yet another level where images are used taken from a helicopter flying around the scene. One can hardly imagine a setup with less control on the recorded image data. The contrast with the very controlled laboratory circumstances of typical IBR systems gets quite outspoken. To obtain realistic image-blending, several pictures are needed and not just the calibration of whatever views have been taken, a sufficiently dense set of pictures are needed. Using enough pictures will also ensure that all areas of the site that are visible from any requested view can indeed be rendered. That is why wide-baseline matching techniques are not an option in this case. An approach for finding the available pictures taken from neighboring viewpoints for a selected viewpoint has been implemented. Traditionally, IBR assumes that the input pictures come in a pre-ordered fashion. This is hardly acceptable when images have been taken with a handheld camera.

Our experience with unstructured lumigraphs was the starting point to generate the new pictures with IBR. But for the work reported here, our existing IBR framework has been combined with the strengths of several other IBR approaches. For instance, dense depth reconstructions can be used as proxies for each camera view. These are now stored

Figure 4: *The helicopter from where the photographs were shot.*

using adaptive hierarchical piece-wise linear approximations , allowing for fast access and display. Most calculations have furthermore been implemented on the graphics card using the latest OpenGL shading language. Unless IBR techniques are allowed to use a huge number of images, sampling the viewing space very densely, they not only need the camera calibration for all images, but also a geometric proxy of the scene. Previously structure and motion techniques have been used to obtain this information. Unfortunately, due to the dominantly planar nature of the site, these techniques would certainly fail. On the other hand the scene is not perfectly planar enough either to simply use planar homographies to describe the geometric relations between the images. Specialized algorithms have been implemented that can overcome this planar degeneracy taking into account some information on the intrinsic parameters of the camera.

4. Acquisition of Imagery

The Ename TimeLine was created with views on a set of geo-referenced virtual models of the complete village. If we want the real component of the movie to seamlessly match the virtual images, we need images of the real site that are recorded from approximately the same positions. For every time instance, there are 36 images, evenly distributed over a circle with a radius of 260 meter. The virtual cameras are positioned 150 meter high and looking towards the center of the circle on the ground. Fig. 1 shows this setup. The easiest way to record images in the real world, taken approximately on this same circle, is by using a helicopter.

On a day with perfect weather conditions in June 2004, the helicopter (fig. 4) arrived at the site. The helicopter pilot was asked to fly the white circle (with a radius of 260 m) at an altitude of 150 meter as shown on the ortho-photo of fig. 5. An additional navigator made sure the pilot could follow these requirements safely. While the helicopter followed this path as precisely as possible, a photographer – as a third crew member – used a digital photo camera to record the scene, focusing his camera all the time on the center of the circle, somewhere in the middle of the archaeological site. As the helicopter needs a certain speed to fly the circle and

Figure 5: *The navigator of the helicopter was asked to follow the flight path shown in this figure.*

Figure 6: *Some of the 130 images taken during the helicopter flight.*

the digital camera needs a certain time to store the high resolution image, an average of 15 images were taken per tour, so the consecutive images were quite far apart. Therefore, the helicopter flew the circle several times at slightly different altitudes and images were taken constantly. Fig. 6 shows some of the 130 digital images that were acquired during the helicopter flight.

5. Grouping of Images

The images that were acquired as described in section 4 show the site from different angles. Consecutive images can be far apart and are therefore not always suited for Structure and Motion (SaM) algorithms. Since the images were recorded during multiple fly-bys, the same content reappears regularly. As explained in section 2, it is not our goal to calibrate the cameras of all images or of the entire circle in one SaM process (rather several, smaller IBR problems are

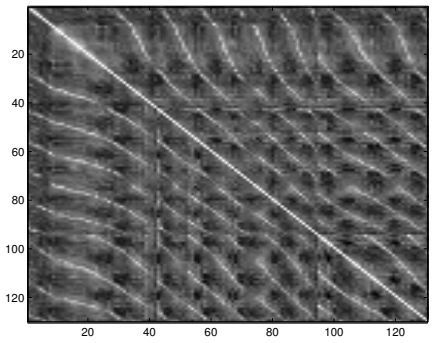

Figure 7: *Matrix with NCC values. White points represent a high correlation, black points mean that there is no correlation between the two images given by the x and y coordinate in the matrix. This matrix can be used to determine a set of images that were taken close to a given image.*

solved). We want to combine different images taken from viewpoints that are quite close to a target viewpoint and apply SaM to this group of images. The camera range spanned by the selected group of images is large enough to solve the IBR problem for the targeted viewpoint and some of its neighbors. It turned out that an average number of four viewpoints could be rendered for every group of images. The process of selecting the group of images for a target viewpoint was therefore repeated nine times. The major task in the grouping process consists in detecting images that were taken close to each other. Manual selection in the 130 images is, while possible, a very tedious task. A simple but effective algorithm has been implemented which does the job.

The algorithm boils down to a comparison of all pairs of images. The center part of one image is selected. An area with the same size is extracted from the other image and slides over this image. For every position, both windows are compared using Normalized Cross Correlation. The highest correlation gives us a measure of how well these images resemble each other. The images are subsampled to decrease processing time. Fig. 7 shows the resulting, symmetric matrix of these values. White points represent a high correlation, black points mean that there is no correlation between the two images given by the x and y coordinate in the matrix. The diagonal is of course perfectly white. One can clearly see the different passes of the helicopter appearing as white lines, connecting consecutive images. The selection process becomes a much easier task now. If we want to reconstruct the cameras and 3D scene from a certain viewpoint, all we need to do is select one image that is taken near the viewpoint we want. The matrix of fig. 7 then lists the best candidates to match this image with.

6. Structure and Motion and Dense 3D

The ultimate goal of the pipeline described in section 2 is to generate new images from specific viewpoints. We have chosen an image- based rendering approach to do so. As will be explained in section 8, this approach produces a kind of high quality interpolation between a set of images recorded in the vicinity of the required viewpoint. As mentioned earlier, our version of this approach also extracts the calibration parameters (both extrinsic and intrinsic) of the camera views and the 3D geometry of the scene – i.e. the scene 'Structure' and camera 'Motion' – and does so directly from the same set of images.

In the last decade tremendous progress has been made in the computer vision community on solving the problem of structure and motion recovery from uncalibrated image sequences. It is beyond the scope of this paper to go into the details of this problem and its solutions. A short overview will be given in paragraph 6.1. Paragraph 6.2 will deal with one specific aspect that was important in this work, namely how to deal with the degenerate case of a nearly planar scene. Some results of the structure and motion recovery are shown in paragraph 6.3

6.1. Structure and Motion Overview

The general problem of structure and motion recovery can be stated as: *Given a set of images or a video sequence, compute the intrinsic and extrinsic camera calibration for all views and the 3D reconstruction of the scene that is visible in these views.* A possible solution to this problem that computes the result completely automatically has been developed in the ESAT-PSI lab [PVV*04] and comprises the following steps:

- First the images are pairwise related to each other. Feature points are extracted in every image and matches are found between consecutive images by comparing these features using a comparison function like Normalized Cross Correlation
- For every consecutive pair of images the epipolar geometry is computed from the previously computed matches. Outliers (which normally correspond to wrong matches) are detected and removed with the help of the epipolar geometry.
- The 3D structure of the scene and the calibration of the camera is initialized for the best suited pair of images. All other cameras are consecutively computed in the same coordinate frame and the 3D pose of the feature points is computed. The resulting reconstruction is valid up to any projective transformation.
- The projective reconstruction is upgraded to metric (Euclidean up to scale) using self-calibration and a bundle adjustment procedure minimizes the total reprojection error of all points in all cameras.
- Since the calibration is now computed, the cameras can be rectified and a dense 3D reconstruction of the scene

can be computed, using stereo and a fusion algorithm that combines information from all stereo pairs into a depth map for every original view.

6.2. Planar Degeneracy

Standard structure and motion algorithms, as described in the previous paragraph, first relate all images in a projective frame and then upgrade this frame through self-calibration. Unfortunately this approach suffers from the existence of critical motions [KTA00] and surfaces [Hof53]. This means that there are cases for which sequences are recorded in a specific fashion such that multiple solutions to the self-calibration procedure exist. Since only one of these solutions corresponds to the real world and is the one we look for, and since this solution can not be distinguished from the other solutions, the result of the pipeline of paragraph 6.1 will in general not be correct. Examples of critical motion sequences are a motion on a line or a circle. A typical example of a critical surface sequence is one in which only a single plane is visible. The archaeological site of Ename is not exactly planar but when viewed from a distance of about 300 meter there is hardly any 3D information outside the ground plane, which will certainly cause problems for standard structure and motion.

In recent years different solutions have been proposed to deal with planar structures. They vary from merely surviving the plane by detecting when only a plane is visible in the images, calibrating and reconstructing in parts where more than this plane is visible and extending this structure to the planar part [PVG02], to effectively dealing with the planar structure (or other critical motions and surfaces) by taking into account more information about the camera intrinsic parameters than in the general structure and motion algorithm [Nis03].

The first approach will not help us in this case because it needs some cameras of the sequence to observe more than a plane which is not the case in the recordings from the helicopter. Therefore the second approach has been implemented. Since it assumes the intrinsics of the camera to be known, one can estimate the essential matrix instead of the fundamental matrix when relating two views. This essential matrix takes into account the intrinsic camera parameters and, unlike the fundamental matrix, is unique for all practical cases, even if all observed points lie in a single plane. Pose estimation of the cameras (the third step of paragraph 6.1) can then be done in a metric frame which alleviates the problems we had with general structure and motion. In order to retrieve the intrinsic parameters we recorded some other, non planar scenes with the camera that was used in the helicopter, using the same settings. These sequences could be processed with our normal algorithms. The calibration we used for the planar scene was inferred from the resulting camera calibration of these extra sequences.

Figure 8: *Four of the seven images that served as input for one structure and motion process. The images were taken during different passes of the helicopter and are selected automatically using the matrix in fig. 7.*

6.3. Structure and Motion Results

For the automatic structure and motion algorithm to succeed, the input images should not differ too much. Techniques exist to deal with wide-baseline cases but these are not applicable here since the result of the structure and motion pipeline will be used by the image-based rendering process of section 8 which needs images with viewpoints that are close. Fig. 8 shows four of seven images in a group used for one of the 9 structure and motion computations mentioned in section 5. The images are selected by choosing one manually and retrieving the six best matching views from the matrix in fig. 7.

The resulting point reconstruction and camera calibration can be seen in Fig. 9. A top and side view are shown. It is clear that the structure of the scene is very close to planar. Fig. 10 shows some views of the dense 3D model that was reconstructed from the images. The dense 3D is far from perfect, due to large homogeneous areas like the fields or due to the large distance to the cameras, like the houses in the background. The quality might not be good enough for ordinary, geometry-based rendering but is sufficient for image- based rendering.

7. Registration

At this stage in the pipeline of section 2 we have available separate sets of images – on average one set per four target views – with corresponding 3D reconstructions and camera calibrations. We want to employ these results to create synthetic viewpoints that coincide with the corresponding target viewpoints. Unfortunately every 3D reconstruction has been computed in its own metric frame. So far, each reconstruction is related to the site coordinate system of fig. 1 via

Figure 9: *The camera calibration of the images and a set of 3D points, representing the scene, retrieved by our structure and motion algorithm. Both a top and a side view are shown. The planar nature of the scene can clearly be recognized.*

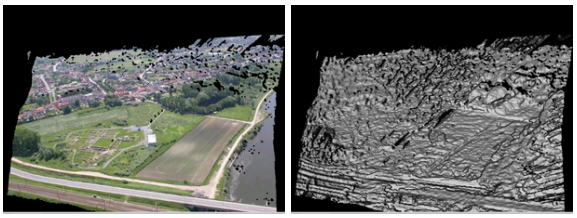

Figure 10: *Dense reconstruction of the scene. The quality is not perfect, certainly in homogeneous areas or in areas that are far away.*

Figure 11: *Pose estimation with manual interaction. Corresponding points (red) are indicated both on the ortho-photo (top) and on the image taken from the helicopter (bottom).*

an unknown metric transformation consisting of a rotation, a translation and a scale factor. If we want to use the reconstruction for image-based rendering, we need to compute these metric transformations. This process is called registration. For clarity we call the coordinate frame of the structure and motion reconstruction the *SaM-frame* and the coordinate frame of fig. 1 the *target frame*.

7.1. General Strategy

A possible strategy to find the metric transformation described above consists of the following two consecutive steps

1. Bring one camera of the structure and motion reconstruction into the target frame, effectively computing the rotational and translational part of the transformation.
2. Compute the scale factor between the two frames from one or more indicated distances.

Step one boils down to a pose estimation process of one camera that has been computed in the SaM-frame in the target frame. For pose estimation, one needs a set of 3D points, with coordinates in the target coordinate system and a set of corresponding 2D points in the image for which the camera pose needs to be computed. If the intrinsic parameters of the camera are known, three point correspondences suffice and it is possible to compute the solution directly, as was shown in the mid 19th century by Grunert [Gru41]. If more than three correspondences are available, a better approach is to compute an initial solution with Grunert's algorithm and then minimize the total reprojection error of all available points with a non-linear optimization step.

Since step one only computes the rotation and translation between the SaM-frame and the target frame, we still need to determine the scale factor between them. This is the second step in the registration process. If two 3D points are known in both coordinate systems, the ratio of both distances is the scale factor we are looking for. One of the points could be the center of the camera that has been computed with pose-estimation, since we already know this in both frames. We just need one more 3D point with known coordinates in both frames. Of course if more points are known, the scale factor can be computed more accurately.

7.2. Interactive Registration

To get at the required 3D-2D correspondences for the pose-estimation step, we ask the user (in our case a member of the

Ename archaeological team) for some manual help. First the 3D points are computed. We have a geo-referenced ortho-photo of the archaeological site at our disposal. Since the remains of the buildings are all approximately situated in the same horizontal plane, a 2D position, indicated by the operator in this ortho-photo, immediately translates to a 3D position in the target frame. For pose-estimation we not only need 3D points but the corresponding 2D points in the image as well. The operator is asked again to indicate these points. Fig. 11 shows such a process. In the top image the relevant part of the ortho-photo is shown. The bottom part shows the image taken from the helicopter. The labeled indicated points are shown superimposed. The scale factor of the second step can be computed from any of the indicated points of which we know the 3D position in both frames. The coordinates in the target frame are known from the ortho-photo and the position in the SaM-frame can be found through a lookup in the corresponding depth map that was computed in section 6.

8. Image-Based Rendering

Image-Based Rendering (IBR) generates novel views by interpolating information from images that are close to the requested view. The advantage over traditional 3D scene model-based rendering is that an exact geometrical description of the scene is not necessary in this approach [LH96] although approximate geometrical information can be used to improve the results [GGSC96, CTCS00]. In the ESAT-PSI lab we have developed an image-based rendering pipeline [HKP*99, KPH*99], and have recently extended it further to incorporate elements from the work of different authors, such as Buehler [CML*01], Pajarola [PSM03] and Evers-Senne [ESK03]. This extended pipeline is used to create each of the 36 target views of the present site (i.e. the last row of the TimeLine matrix), each time exploiting a set of relevant, grouped images (step 2).

In contrast to the first Lumigraphs, the ESAT-PSI pipeline can deal with an unstructured set of images. All source images have their associated camera pose estimates, which we have computed using Structure and Motion and which were registered with the target frame. The algorithm behaves like an extension of view-dependent texture mapping and uses depth maps generated through SaM as an approximation of the 3D geometry [KPH*99]. Not all cameras that we have at our disposal are equally suited for rendering from a certain virtual viewpoint and therefore a set of cameras is selected. In our implementation the cameras which will be used to render the virtual view are selected in a similar fashion as described in [ESK03]. During operation the user can set the number of selected cameras, the relative importance of the selection criteria, etc.

Rather than using a fixed 3D geometry, Pajarola et al. [PSM03] propose a technique to approximate the 3D information in a non-uniform way. For every depth map a restricted quadtree is built. This quadtree holds the hierarchical representation of the depth map and has the additional property that, no matter which hierarchical level is chosen, the topology of the scene is preserved. The construction of this quadtree can be done beforehand in an off-line step, reducing the on-line processing time significantly.

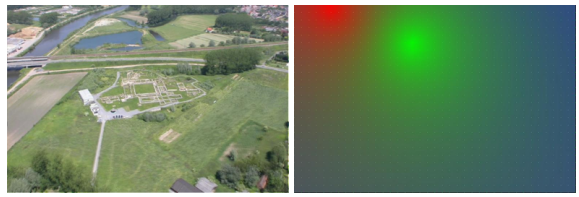

Figure 12: *A virtual image, rendered from three helicopter images (left). The corresponding blendfield (right) clearly shows the relative importance of each camera.*

The key property of image-based rendering techniques is the combination of different images to render a new virtual view. Not every image that is used for the rendering has the same impact in all areas of the new image. It is desired that the importance of each image depends on the quality of its camera, its 3D reconstruction, ... evaluated at every pixel of the virtual view. To this end a *blendfield* is constructed. This blendfield describes the relative importance of each camera for every pixel in the virtual view. In order to generate the resulting image, each selected camera projects its own image onto its own quadtree and is rendered from the requested viewpoint. All renderings are combined into one image using the weights in the blendfield.

Although the 36 necessary views can be rendered off-line, one can easily think of other applications where interactivity is important. That is why we have implemented as many operations as possible on the GPU. Fig. 12 shows a virtual view, reconstructed from three selected cameras with its corresponding blendfield. The relative importance of each image at each pixel of the virtual view is given by the value of the blendfield at this pixel. For visualization of this blendfield, a color is assigned to each of the three images used to generate this virtual view: red, green and blue. A reddish pixel in the right picture of fig. 12 for example denotes a pixel where the influence of the first image in the final rendered image is high.

8.1. Refining cameras

During processing it became apparent that the calibration of the cameras sometimes needed improvement. For the Structure and Motion process to deliver accurate results feature points need to be tracked throughout the entire sequence of images. The longer the feature tracks, the better the result will be. Unfortunately in this specific case images contain many homogeneous areas, like the fields and the river. These

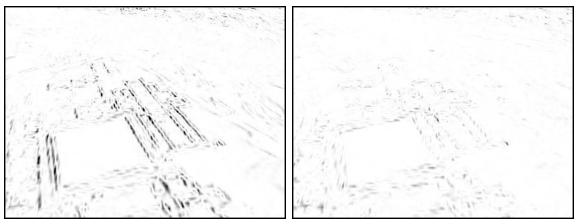

Figure 13: *The difference image between the reference and one other camera before (left) and after (right) minimization. Black pixels indicate a high, white pixels a low intensity difference. The overall error is reduced by a factor of 2.4.*

areas are a nightmare for feature detectors. A refinement process has been implemented that improves the calibration result. One camera and its corresponding 3D information is selected as the reference and a virtual viewpoint is chosen. All other images are sequentially projected through their corresponding camera via the reference 3D scene onto the virtual view. The difference between this rendering and the one obtained from the reference view is computed. A non-linear optimization algorithm aims at minimizing this difference by allowing each camera to adjust its position and orientation. Fig. 13 shows the difference before and after the optimization for one camera. The resulting camera calibration is more accurate than the original because also the homogeneous areas will have their impact. All projection and difference operations are performed on the GPU, speeding up this minimization process significantly.

9. Results

The pipeline of section 2 was executed for 9 sets of helicopter images. The reconstructions of these sets allowed us to generate the necessary 36 views for the QTVR object movie, yielding an average of four newly generated images per set of helicopter images. Fig. 14 shows four of these new views, evenly distributed along the circle.

The new images fit very well with the virtual images of the TimeLine. This can be evaluated by comparing the position of the remaining foundations of the buildings with the walls of the virtual images. Fig. 15 shows the last two time periods in the updated TimeLine. The first image is a virtual reconstruction, representing the site as it was around 1780. The second is a photograph of today's situation, generated with the IBR pipeline. Fig. 16 shows two viewpoints where the virtual reconstruction of 1780 has been merged with the newly generated images. The buildings of the archaeological site clearly correspond to the foundations in the real images. The area in the background, including the church (visible in 15), does not fit so well. This can easily be explained by errors in the registration step. The reference data in this step consisted of an ortho-photo of the archaeological site. This data corresponds to an area in the helicopter images of about

Figure 14: *Four of the newly generated images of the archaeological site*

Figure 15: *The newly rendered image for position 10 and its virtual counterpart. The fit is very good in the area of the archaeological site. The areas near the border of the image, like the church, have a larger error.*

25 percent of the total image. The registration is therefore good for this area but the error in other image parts increases rapidly.

10. Acknowledgments

The authors thank the Fund for Scientific Research of Flanders (FWO-Vlaanderen) and the European IST projects *IN-*

Figure 16: *Two viewpoints of the time period around 1780 have been merged with the corresponding image-based rendered images of the current situation. The fit is remarkably good.*

VIEW and *EPOCH* for their support, as well as Noel Colman, aerial photographer, for providing the helicopter service and for his advice.

References

[App] APPLE DEVELOPER C.: *QuickTime VR*. http://developer.apple.com/documentation/-QuickTime/InsideQT_QTVR/index.html.

[CML*01] C. B., M. B., L. M., S. G., M. C.: Unstructured lumigraph rendering. *Proc, SIGGRAPH 2001* (2001), 425 – 432.

[CTCS00] CHAI J.-X., TONG X., CHAN S.-C., SHUM H.-Y.: Plenoptic sampling. *Proc. SIGGRAPH 2000* (2000), 307–318.

[ESK03] EVERS-SENNE J.-F., KOCH R.: Image based interactive rendering with view dependent geometry. In *Proc. Eurographics 2003* (2003), Computer Graphics Forum, Eurographics Association, pp. 573–582.

[GGSC96] GORTLER S., GRZESZCZUK R., SZELISKI R., COHEN M. F.: The lumigraph. *Proc. SIGGRAPH 1996* (1996), 43–54.

[Gru41] GRUNERT J.: Das pothenotische problem in erweiterter gestalt nebst ueber seine anwendungen in der geodäsie. *Grunerts Archiv für Mathematik und Physik, Band 1* (1841), 238–248.

[HKP*99] HEIGL B., KOCH R., POLLEFEYS M., DENZLER J., GOOL L. V.: Plenoptic modeling and rendering from image sequences taken by hand-held camera. In *DAGM'99* (1999), pp. 94–101.

[Hof53] HOFMANN W.: *Das problem der Gefährlichen Flächen in Theorie und Praxis -Ein Beitrag zur Hauptaufgabe der Photogrammetrie*. PhD thesis, Fakultät für Bauwesen, Technische Universität München, Germany, 1953.

[KPH*99] KOCH R., POLLEFEYS M., HEIGL B., VAN-GOOL L. J., NIEMANN H.: Calibration of hand-held camera sequences for plenoptic modeling. In *ICCV (1)* (1999), pp. 585–591.

[KTA00] KAHL F., TRIGGS B., AASTROM K.: Critical motions for auto-calibration when some intrinsic parameters can vary. *Journal of Mathematical Imaging and Vision 13*, 2 (October 2000).

[LH96] LEVOY M., HANRAHAN P.: Lightfield rendering. *Proc. SIGGRAPH 1996* (1996), 31–42.

[Nis03] NISTER D.: An efficient solution to the five-point relative pose problem. In *Proc. Conf. Computer Vision and Pattern Recognition* (2003), vol. 2, IEEE Computer Society Press, pp. 195–202.

[PSC02] PLETINCKX D., SILBERMAN N., CALLEBAUT D.: The saint laurentius church in ename and its role in the francia media heritage initiative. In *Proc. VAST 2001, Virtual Reality, Arachaeology and Cultural Heritage* (2002), ACM Siggraph, pp. 197–204.

[PSM03] PAJAROLA R., SAINZ M., MENG Y.: *Depth-Mesh Objects: Fast Depth-Image Meshing and Warping*. Tech. Rep. 03-02, 2003.

[PVG02] POLLEFEYS M., VERBIEST F., GOOL L. J. V.: Surviving dominant planes in uncalibrated structure and motion recovery. In *Proceedings of the 7th European Conference on Computer Vision-Part II* (2002), pp. 837–851.

[PVV*04] POLLEFEYS M., VANGOOL L., VERGAUWEN M., VERBIEST F., CORNELIS K., TOPS J., KOCH R.: Visual modeling with a hand-held camera. *International Journal of Computer Vision 59*, 3 (2004), 207–232.